基于 MLP 神经网络的分组密码算法能量分析研究 *

王　恺[1,2]，蔡爵嵩[1]，严迎建[1]

(1. 战略支援部队信息工程大学，郑州 450001; 2. 中国人民解放军 32125 部队，济南 250001)

摘　要：随着嵌入式密码设备的广泛应用，侧信道分析(SCA)成为其安全威胁之一。通过对密码算法物理实现过程中的泄漏信息进行分析实现密钥恢复，进而对密码算法实现的安全性进行评估。多层感知器(MLP)是一种人工神经网络结构，为了精简用于能量分析的 MLP 网络结构，减少模型的训练参数和训练时间，针对基于汉明重量(HW)和基于比特的 MLP 神经网络的模型进行了研究，输出类别由 256 分类分别减少为 9 分类和 2 分类。通过采集 AES 密码算法运行过程中的能量曲线，对所提出的 MLP 神经网络进行训练和测试。实验结果表明，该模型在确保预测精度的前提下，能减少 MLP 神经网络 84%的训练参数和 28%的训练时间，并减少了密钥恢复阶段需要的能量曲线数量，最少只需要 1 条能量曲线，即可完成 AES 算法完整密钥的恢复。实验验证了模型的有效性，使用该模型可以对分组密码算法实现的安全性进行分析和评估。

关键词：侧信道分析；深度学习；MLP；密码芯片；AES

中图分类号：TP309.1　　doi: 10.19734/j.issn.1001-3695.2019.12.0703

Research on side channel analysis of block cipher algorithm based on MLP neural network

Wang Kai[1, 2], Cai Juesong[1], Yan Yingjian[1]

(1. *Strategic Support Force Information Engineering University*, *Zhengzhou* 450001, *China*; 2. 32125 *Troops of PLA*, *Jinan* 250001, *China*)

Abstract: With the widespread application of embedded cryptographic equipment, side channel analysis (SCA) has become one of its security threats. The key information is recovered by analyzing the leaked information during the physical implementation of the cryptographic algorithm. Furthermore, the security of the cryptographic algorithm can be evaluated. MLP is an artificial neural network structure. In order to streamline the MLP network structure for energy analysis and reduce the training parameters and training time of the model, this paper studied the models based on Hamming weight (HW) and bit-based neural networks, and the output categories were reduced from 256 to 9 and 2 respectively. The power trace during the operation of the AES cryptographic algorithm was collected through experiments. This paper trained and tested the proposed MLP neural network. The results show that the model can reduce the training parameters of the MLP neural network by 84% and the training time by 28%, and reduce the number of power traces required during the key recovery phase, while ensuring the prediction accuracy. At least only one power trace is needed to complete the recovery of the AES algorithm's complete key. The validity of the model is verified by experiments, and the security of the block cipher algorithm can be analyzed and evaluated by using the model.

Key words: side channel analysis; deep learning; multi-layer perceptron (MLP); cryptographic chips; AES

0　引言

嵌入式设备如智能卡、RFID 射频标签和各种物联网设备在本文的生活中得到了广泛应用，这些设备使用密码算法实现加解密操作，从而保护数据的安全。但是，在密码算法的执行过程中，设备所处理秘密信息会在能量消耗[1]、处理时间[2]、电磁辐射[3]等方面泄漏，泄漏的信息依赖于所处理的数据和执行的操作，因此可以通过对泄漏信息进行分析来恢复敏感信息。

侧信道分析是由 Kocher 于 1996 年首次提出的[2]，他通过对时间序列分析进行密钥恢复，随后提出了更强大和更通用的分析形式[1]，称为差分能量分析(DPA)。后来，Mulder 等人将差分分析技术应用到差分电磁分析(DEMA)[4,5]和其他密码算法如 ECC 或 RSA 等。除了简单和差分分析分析外，模板攻击被认为是最有效的分析方式，它假设密码分析者可以获得一个相同的目标设备，并对目标设备完全可控，利用泄漏信号对随机变量统计特性进行建模，用判别分析的方法获取目标设备泄漏信息中所隐藏的秘密信息。

近年来，密码学界探索了基于机器学习和深度学习的密码分析新方法，这对加密算法实现的安全性造成了威胁。刘飚等人[6]使用支持向量机(SVM)方法对电磁泄漏信息进行分析，从而实现密钥的恢复。Lerman 等人[7]通过实验验证了 SVM 算法可以对带掩码防护的 AES 算法实现密钥恢复。

作为机器学习的一个分支，深度学习使用深度神经网络从复杂数据中学习特征，并对另一组数据分析作出决策，具有良好的特征提取和分类功能，已成为当前的研究热点，许多研究已经证明了深度神经网络在 SCA 中的性能。Maghrebi 等人[8]最早研究了深度学习在密码算法实现中的应用，将多层感知器(MLP)和卷积神经网络(CNN)等深度学习模型应用于 SCA。在文献[9]中，作者提出了一种基于神经网络的侧信道分析方法，对 DPA contest V4 的带掩码防护 AES 算法实现破解。文献[10]评估卷积神经网络(CNN)的在处理带防护措施

收稿日期：2019-12-04；修回日期：2020-01-16　基金项目：军队科研资助项目

作者简介：王恺(1990-)，男，山西昔阳人，助理工程师，硕士研究生，主要研究方向为安全专用芯片设计技术、深度学习(yixiwk@163.com)；蔡爵嵩(1992-)，男，四川绵阳人，硕士研究生，主要研究方向为安全专用芯片设计技术、深度学习；严迎建(1973-).男，河南扶沟人，教授，博导，博士，主要研究方向为安全专用芯片设计技术等.

或未对齐曲线条件下性能。Benadjila 等人[11]通过实验给出了 MLP 模型的超参数选择的方案，进一步证明了深度学习在模板型 SCA 中的强大功能。

本文对文献[11]中提出的 MLP 神经网络模型结构进行了更为深入的研究，针对模型参数较多、各层连接关系复杂和训练时间较长的缺点，对模型的结构进行优化和精简，输出层由 256 分类改为 9 分类和 2 分类，本文提出了基于汉明重量的 MLP 模型(HW-MLP)和基于比特的 MLP 模型(bit-MLP)。实验结果表明，相比于文献[11]中提出的 256 分类模型，本文提出的 2 种模型在确保预测精度的前提下，减少了 84%的训练参数和 28%的训练时间，在密钥恢复阶段最少只需要 1 条能量曲线，即可完成 AES 加密算法密钥的恢复。本文也通过实验对该模型的防护策略进行研究，使用该模型可以有效地对分组密码算法实现的安全性进行分析和评估。

1　密码算法能量分析和 MLP 神经网络

1.1　模板型能量分析

模板型(Profiled)能量分析假设如下：假设对手有两台完全相同的密码设备，一台建模设备和一台目标设备。分析者能够完全控制建模设备的输入和输出，能够通过技术手段非常精确地采集和刻画设备工作时的能量信息；目标设备运行的是具有未知密钥 $k^*\in K$ 的加密算法，分析的目标是恢复密钥字节 k^*。因此模板型的侧信道分析分两个阶段执行：建模阶段和分析阶段，对应于深度学习中的训练阶段和测试阶段。

1) 建模阶段(profiling phase)

从建模设备中采集能量曲线，使用建模设备与密钥相关泄漏信息来构建特定的泄漏模型。分析者对每个可能的 $k\in K$，分别采集 N 条能量曲线 $P_k=\{T_i(k)|i=1,\cdots,N\}$ 构成训练集合 X：

$$X=\bigcup_{k=0}^{255}P_k \tag{1}$$

计算概率分布函数 ξ 的估计值 e_k：

$$e_k=\xi[T=t\,|\,(P,K)=(p,k)] \tag{2}$$

其中 T 表示所采集的能量曲线数据集，在已知明文 p_i 和密钥 k_i 下获得能量曲线 t_i，估计值 e_k 是从训练集 $D_p=\{t_i, p_i, k_i\}$，$i=1,\cdots,S_p$ 中计算出来的。

2) 分析阶段(Analysis phase)

从目标设备采集能量曲线并根据泄漏信息进行分类，测试集 $D_a=\{t_i, p_i\}$, $i=1,\cdots,S_a$ 用于恢复正确的密钥，最终目标是通过能量曲线的测试集恢复密钥 k^*，概率分布函数估计值 e_k 最高值。因此，分析者构建模型必须能正确区分概率分布函数 ξ 的估计值 e_k, $k\in K$，实际通过最大似然函数对估计值 e_k 进行计算，对每个可能的猜测密钥 $k\in K$ 计算最大似然估计 $L(K)$：

$$\log L[k]=\sum_{i=1}^{S_a}\log\xi[(P,K)=(p,k)\,|\,T=t_i] \tag{3}$$

其中，$L(K)$是对应于猜测密钥 k 的对数似然概率，从目标设备中采集到的能量曲线使 $L(K)$最大的值对应的 $k\in K$ 为正确的密钥值。

1.2　非模板型能量分析

非模板型(Non-Profiled)能量分析使用统计分析来检测泄漏信息和敏感变量之间的相关性。密码算法能量分析利用了这样一个事实：密码设备的瞬时能量消耗依赖于设备所处理的数据和设备所执行的操作。非模板型分析方法主要包括简单能量分析(Simple Power Analysis, SPA)、差分能量分析(Different Power Analysis, DPA)、相关能量分析(Correlation Power Analysis, CPA)等[12]。这种类型的 SCA 对应于假设较弱的分析者，他们只能访问在目标设备上捕获的物理泄漏，用于恢复密钥。

1.3　MLP 神经网络

多层感知器(Multi-Layer Perceptron, MLP)是一种由多个感知器单元组成的神经网络[14,15]，如图 1 所示。每一层的所有感知器与下一层的所有感知器相连。MLP 由输入层、输出层和一系列中间层(隐藏层)组成。每一层由一个或多个感知器单元组成。MLP 的权重值和偏差值是随机梯度下降过程中更新的可训练参数。

本文主要研究多层感知器(MLP)神经网络，它由多个线性函数和非线性激活函数组成的函数 F 相联系，该函数具有计算效率高、导数有界且易于求导的特点。综上所述，可以将 MLP 表示为

$$F(x)=s\circ\lambda_n\circ\sigma_{n-1}\circ\lambda_{n-1}\circ\cdots\circ\lambda_1(x)=y \tag{4}$$

其中 λ_n 是全连接层，σ_i 为激活函数，s 为 Softmax 函数。

MLP 神经网络是由神经元形成一个网格状的结构，该结构被分成多个连接层，每个神经元的值为

$$n_{i,j}=f\left(\sum_j n_{i-1,j}*w_j+b_j\right) \tag{5}$$

其中 w 为相邻层之间神经元连接权重值，b 为该神经元的偏置值，f 为激活函数，当前层的神经元都是上一层中每个相连神经元的输出值的函数。常用的激活函数包括 ReLU、Sigmoid、Tanh 和 Softmax。

MLP 网络模型算法核心思想是通过前向传播得到误差，再把误差通过反向传播实现权重值 w 的修正，最终得到最优模型。在反向传播过程中通常使用随机梯度下降法对权重值进行修正，梯度下降法的原理是计算损失函数关于所有内部变量的梯度，并进行反向传播。内部变量通常是权重值，根据损失函数所跨越曲面的最陡下降方向进行调整[16]。

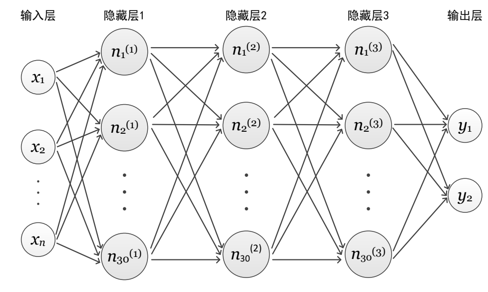


图 1　具有 3 个隐藏层的 MLP 神经网络

Fig. 1　MLP neural network with 3 hidden layers

神经网络是一种计算开销非常大的技术，与传统分析方式 DPA、CPA 相比，它需要花费更多的时间和内存资源。但是，它对于侧信道能量分析具有以下明显的优势[17]：

a) 不需要手动选择特征点，卷积层和全连接层通过自动特征提取可以识别和提取与曲线相关的特征；

b) 卷积层可以独立于其在数据中的位置来提取特征，因此深度神经网络能够对抗非稳定的时钟周期和消除随机延迟策略带来的抖动；

c) 由于深度神经网络是一个高度参数化的模型，因此可以基于超参数优化来优化分类精度，进而提升侧信道分析的成功率；

d) 深度神经网络可以实现高度复杂的功能，对广泛采用的防护策略(如掩码策略或混乱策略)有一定的分析能力。

1.4　汉明重量模型

汉明重量模型是计算寄存器在某一时刻所存储的数据 1 个数，根据 1 的个数来刻画寄存器的能量消耗[12]。对一个 k 位二进制数据的汉明重量模型的形式化表述为

$$HW(V)=\sum_{i=0}^{k-1}v_i \tag{6}$$

其中 $V=(v_{k-1}\cdots v_1v_0)(v_i=0 \text{ or } 1)$，此模型通常用于预充电模式的设备中能量消耗模型的刻画，即在数据变化之前，寄存器中所有编码置为 0 或 1。

能量消耗模型是对实际能量消耗的一种模拟，模拟精度

越高则分析者恢复出密码设备所使用的敏感信息或密钥信息的能力就越强。汉明重量模型可以在分析者对密码芯片的网表一无所知或仅知道很少一部分的情况下进行分析。在汉明重量模型中，分析者假设能量消耗与被处理的数据中操作位的比特数存在线性关系，事实上，数据的汉明重量与处理该数据造成的能量消耗并非完全相关，只限于一些具体的场景。在 AES 算法的侧信道分析过程中，通常选取某个 S 盒的输出作为分析点，在该点的输出值的汉明重量通常与泄漏值成线性关系，如图 2 所示，能量曲线的颜色由深到浅说明泄漏值与汉明重量成线性关系。

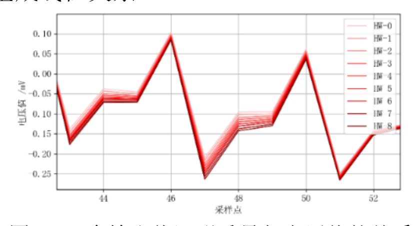


图 2 S 盒输出值汉明重量与电压值的关系

Fig. 2 The relation between output hamming weight and voltage value of S-box

2 基于 MLP 神经网络的能量分析流程和模型

2.1 能量分析流程

基于 MLP 神经网络模型的能量分析总体流程如图 3 所示，具体操作步骤如下：

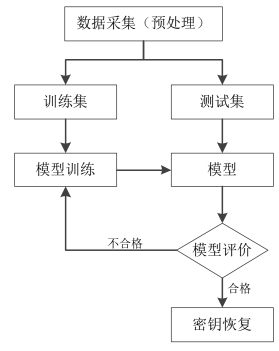


图 3 基于 MLP 神经网络的分析流程

Fig. 3 The analysis process based on MLP neural network

a) 数据采集和预处理。在传统的模板攻击和基于机器学习的 SCA 中，数据预处理步骤实际上是狭义的降维。通常采集到的原始能量数据具有噪声大、维度高的特点，需要对数据进行预处理以降低计算复杂度，因此预处理过程不能省略。但是基于深度学习的 SCA，可以省去降维的预处理步骤，直接对数据集进行预处理。通常将采集到的能量曲线数据分为训练数据集和测试数据集两部分，训练数据集用于模型训练和建模，测试数据集用于模型评价和密钥恢复。

b) 模型训练。使用训练数据对模型进行训练，在训练过程中通常采用交叉验证的方式对模型的训练效果进行评估，并且对不同的深度神经网络或机器学习算法进行对比，在本文实验中对支持向量机算法(SVM)进行对比.

c) 模型评价。通常使用多种评价策略来评价模型的性能，或为参数化模型选择最优参数。在能量分析中主要评价指标有：模型的预测精度、密钥猜测熵、训练时间、恢复密钥所需要的曲线条数、计算资源消耗等。

d) 密钥恢复。在密钥恢复过程中，通常采用“分而治之”的策略，对 AES 算法的 16 个密钥字节逐个进行恢复。对于基于汉明重量的模型，可以通过汉明重量的预测值和已知的明文数据，计算出可能的密钥值，通过使用多条曲线所对应的不同汉明重量和不同的明文数据，通过将猜测值取交集的形式逐步缩小密钥猜测范围，最终获得子密钥。

2.2 分析模型选择

与传统的侧信道分析方法如 DPA、CPA 和模板攻击等类似，基于深度学习的能量分析也需要定义泄漏模型，基于 MLP 神经网络模型的分析是在有监督的条件下进行的，因此，训练阶段和测试阶段的曲线需要根据泄漏模型选择相应的标签值，根据所选标签值的不同，神经网络最终的分类数量也是不同的。对分组密码算法进行分析时，通常选 S 盒输出值为分析点，该点处的操作为非线性变换，能量消耗较大，对于不同数据区分较明显。

如果泄漏模型是 AES 加密算法第一轮运算 S 盒输出值，如图 4 所示，在神经网络输出值的类别数有以下几种：a) 密钥字节身份模型(ID 模型)输出值为 2^8=256 类；b) 汉明重量模型(HW 模型)输出值为 9 类；c) 单比特模型(bit 模型)输出值为 0 或 1，共 2 类。

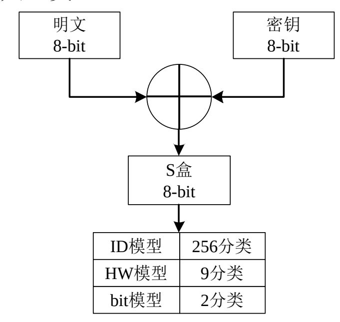


图 4 使用 S 盒输出值构建分析模型

Fig. 4 Use S-box output to build the analysis model

在本文实验中，使用 ID 模型、HW 模型和 bit 模型用于 MLP 神经网络建模，本文分别简称为 ID-MLP 模型、HW-MLP 模型和 bit-HW 模型。

2.3 ID-MLP 模型

设计该模型的目的是将 Benadjila 等人[11]提出的 MLP 模型引入到本文的实验平台中，对其参数进行优化和确定，以达到最佳的分析效果，并将该模型作为参考模型与本文提出的模型进行性能对比。

由于采集到的单条能量曲线为时间序列，因此 MLP 神经网络的输入层节点数为能量曲线的采样点个数，输出层为 256 个节点，输出层使用 Softmax 激活函数，ID-MLP 模型共 6 层。为提高测试精度，使用 tanh 激活函数，模型结构如表 1 所示。在训练过程中，标签值为 S 盒的输出值，共 2^8=256 种，因此在密钥恢复过程中，可以直接对密钥字节进行恢复，通过分而治之的策略对 16 个密钥字节进行恢复。

表 1 ID-MLP 模型结构

Tab. 1 ID-MLP model structure

类型	节点数量	输出大小	激活函数	参数个数
输入层	N	1×N	tanh	
隐藏层 N_1	200	1×200	tanh	200×(N+1)
隐藏层 N_2	200	1×200	tanh	40200
隐藏层 N_3	200	1×200	tanh	40200
隐藏层 N_4	200	1×200	tanh	40200
隐藏层 N_5	200	1×200	tanh	40200
输出层	256	1×256	Softmax	51456

2.4 HW-MLP 模型

在实验中采用随机化网格搜索法获取 MLP 结构的超参数，主要包括：全连接层数 N_{layer}，隐藏层节点数 node，迭代次数 epochs，单次训练数据量 batch_size，学习率和激活函数类型等。具体结构设计如表 2 所示。

与 ID-MLP 相比，HW-MLP 模型减少了隐藏层的层数，减少了各隐藏层节点数，增大了学习率，并使用 tanh 激活函数。在各层的结构设计上，节点数逐层递减，整个网络为倒

梯形结构。通过实验证明该模型的训练精度和测试精度提升明显，训练时间和训练参数大幅减少。

表 2　HW-MLP 模型结构

Tab. 2　HW-MLP model structure

类型	节点数量	输出大小	激活函数	参数个数
输入层	N	1×N	tanh	
隐藏层 N_1	200	1×200	tanh	200×(N+1)
隐藏层 N_2	160	1×160	tanh	32160
隐藏层 N_3	120	1×120	tanh	19320
隐藏层 N_4	80	1×80	tanh	9680
输出层	9	1×9	Softmax	729

通过 HW-MLP 模型最终可以预测得到输入的能量曲线所对应的汉明重量值，通过密钥恢复算法，可以使用少量曲线恢复初始密钥字节，具体算法伪代码如算法 1 所示。该算法不局限于特定的神经网络或机器学习算法类型，可以用于任意基于汉明重量模型的模板型能量分析。

算法 1　通过 HW 值恢复密钥字节算法

```
输入：HW_value (HWi) 1≤i≤N, planitexts(pi) 1≤i≤N。
输出：key。
for i∈N do
  for k∈K do
    HWguess=S_box(k⊕pi)
    if HWguess==HWi do
      set HWguess to the HW_set[i]
  HW_set[i]=HW_set[i] ∩ HW_set[i-1]
  if element number of HW_set[i]==1 do
    key= HW_set[i]
end for
return key
```

2.5　bit-MLP 模型

为了进一步精简网络模型和训练参数，使用 bit 模型进行建模，与 HW 模型相比，输出结果为 2 分类，因此进一步减小网络层数和节点个数，具体结构如表 3 所示。通过为每个密钥字节建立 8 个 bit 模型，可以直接对密钥比特进行恢复，进而恢复完整密钥。

表 3　bit-MLP 模型结构

Tab. 3　Bit-MLP model structure

<table>
<tr><th>类型</th><th>节点数量</th><th>输出大小</th><th>激活函数</th><th>参数个数</th></tr>
<tr><td>输入层</td><td>N</td><td>1×N</td><td>tanh</td><td></td></tr>
<tr><td>隐藏层 N₁</td><td rowspan="4">200</td><td rowspan="4">1×200</td><td rowspan="4">tanh</td><td rowspan="4">200×(N+1)</td></tr>
<tr><td>隐藏层 N2</td></tr>
<tr><td>隐藏层 N3</td></tr>
<tr><td>隐藏层 N4</td></tr>
<tr><td>隐藏层 N₂</td><td rowspan="4">120</td><td rowspan="4">1×120</td><td rowspan="4">tanh</td><td rowspan="4">24120</td></tr>
<tr><td>隐藏层 N2</td></tr>
<tr><td>隐藏层 N3</td></tr>
<tr><td>隐藏层 N4</td></tr>
<tr><td>隐藏层 N₃</td><td rowspan="4">40</td><td rowspan="4">1×40</td><td rowspan="4">tanh</td><td rowspan="4">4840</td></tr>
<tr><td>隐藏层 N2</td></tr>
<tr><td>隐藏层 N3</td></tr>
<tr><td>隐藏层 N4</td></tr>
<tr><td>输出层</td><td>2</td><td>1×2</td><td>Softmax</td><td>82</td></tr>
</table>

3　实验与分析

3.1　数据采集

为确保实验结果真实有效，本文所有实验都采用相同的硬件配置。使用 ChipWhisperer Lite 实验平台采集能量曲线，ChipWhisperer 是一套完整的开源工具链，主要用于侧信道能量分析和故障注入。目标芯片为 XMEGA128D4 单片机，目标密码算法为 AES，分组长度和密钥长度均为 128bit。通过 ChipWhisperer Lite 采集 60000 条能量曲线，使用随机明文和随机密钥进行加密，其中 50000 条能量曲线作为训练集，10000 条能量曲线作为测试集。

单条能量曲线如图 5 所示，利用 AES 算法的知识，能很容易确定第一轮运算中 16 个 S 盒的近似位置，右半部分形状较为规则的区域为 16 个 S 盒所对应的能量曲线。图 6 所示为前 2 个 S 盒能量曲线放大之后的图形，其中最大值或最小值处为其泄漏位置，通过观察和计算可以确定单个 S 盒对应的能量曲线包含 72 个采样点，因此在后续实验中，单条曲线样本输入大小为 1×72。

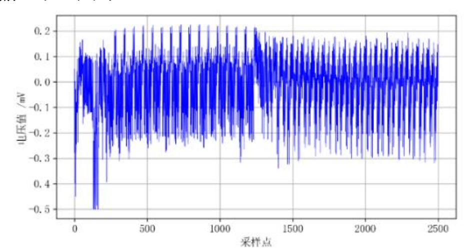


图 5　单条能量曲线波形

Fig. 5　A single power trace waveform

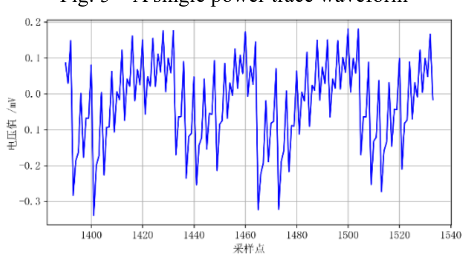


图 6　S 盒能量曲线放大后波形

Fig. 6　The power trace of S-box is magnified

3.2　实验评价指标

通过以下评价指标对实验结果进行分析：

a) 模型精度和损失函数。精度是对神经网络进行训练和评估的最常见的指标，它被定义为在数据集上分类成功的概率，训练精度对应于每个 epoch 结束时得出的最大训练精度，测试精度对应于每个 epoch 结束时得出的最大验证精度。训练精度是训练过程中的一个重要指标，通过观察每个 epoch 训练精度的变化，可以判断训练的模型是否满足神经网络拟合和泛化，训练精度的提高也表明了反向传播算法是否收敛到正确的权重值和偏差值。损失函数是用来评估模型的预测值与真实值的不一致程度，损失函数越小，模型的鲁棒性越好。

b) 恢复密钥字节所需要的曲线条数。通过对比恢复单个密钥字节的曲线条数，对模型的性能进行比较，所需要的曲线条数越少，模型的性能越好。

c) 恢复完整密钥所需要的曲线条数。通过对比恢复全密钥字节的曲线条数，对模型的性能进行比较，所需要的曲线条数越少，模型的性能越好。

d) 模型参数和计算时间。在具有相同预测精度的前提下，模型参数越少、模型结构越精简，模型训练时间花费时间越短，性能越好。

3.3　实验结果

1) ID-MLP 模型实验

使用 ID-MLP 模型对密钥字节进行恢复，以第 0 字节为例进行测试，模型在 epochs=200、batch_size=100 的条件下进行训练，最终训练精度达到 98.00%，平均测试精度达到 98.89%，相比于 Benadjila-MLP 模型，训练精度提升明显，LOSS 值下降较快，恢复密钥所需要的能量曲线更少，如图 7 所示。

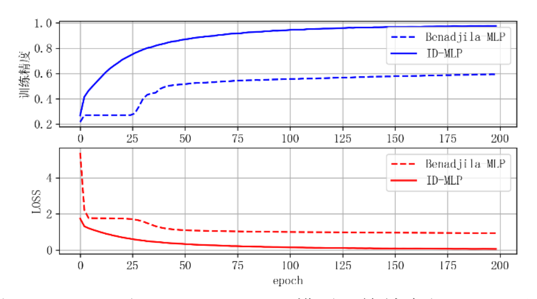


图 7　ID-MLP 与 Benadjila-MLP 模型训练精度与 LOSS 对比

Fig. 7　Comparison of training accuracy and LOSS between ID-MLP and Benadjila-MLP models

恢复其他密钥字节的测试精度和曲线条数如表 4 所示，经计算，平均训练精度达到 97.88%，平均测试精度为 95.51%，恢复密钥字节平均需要 1.050 条能量曲线。

表 4　各 S 盒实验数据和恢复密钥字节需要的能量曲线条数

Tab. 4　Experimental data of each S-box and the number of power trace required to recover key bytes

序号	训练精度/%	训练 LOSS/%	测试精度/%	曲线数量
S_box_00	98.00	7.40	98.89	1.011
S_box_01	97.70	8.40	96.63	1.035
S_box_02	97.40	9.70	95.69	1.045
S_box_03	97.90	7.60	97.22	1.029
S_box_04	97.50	9.10	90.67	1.103
S_box_05	98.10	6.70	95.96	1.042
S_box_06	97.70	8.00	97.20	1.029
S_box_07	98.30	6.10	94.18	1.062
S_box_08	97.90	7.40	95.70	1.045
S_box_09	98.40	5.90	97.94	1.021
S_box_10	97.80	7.90	97.62	1.024
S_box_11	98.30	6.40	97.13	1.030
S_box_12	97.90	7.60	95.53	1.047
S_box_13	98.10	6.60	97.45	1.026
S_box_14	97.50	9.40	96.10	1.041
S_box_15	97.60	8.70	84.26	1.187

对恢复完整密钥共进行 1000 次实验，所需要的能量曲线条数分布柱状图如图 8 所示，平均需要 1.563 条曲线可完成密钥恢复，其中有 487 次实验仅需要 1 条即可完成恢复完整密钥。因此，通过将 Benadjila-MLP 模型进行改进，改入到本文的实验平台，实验数据表明，该模型能达到较好的预测效果。

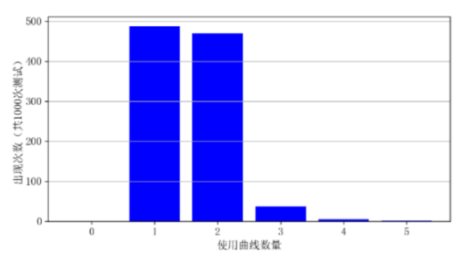


图 8　恢复 128bit 密钥需要的曲线条数分布

Fig. 8　Trace number distribution required to recover 128bit key

1) HW-MLP 模型实验

使用 HW-MLP 模型对密钥字节的汉明重量进行恢复，以第 0 字节密钥为例进行测试，模型在 epochs=200、batch_size=100 的条件下进行训练，随着迭代次数的增加，训练精度不断提升，LOSS 值不断下降，最终训练精度达到 98.00%，测试精度达到 97.11%，如图 9 所示，使用汉明重量密钥恢复算法，恢复第 0 字节密钥平均需要曲线条数为 4.045 条，与 SVM 算法相比精度提升较为明显，所需要的能量曲线更少，训练时间也较短，如表 5 所示。

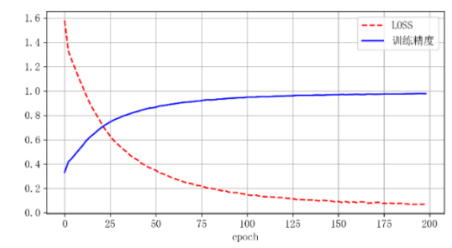


图 9　HW-MLP 模型训练精度与 LOSS

Fig. 9　Training accuracy and LOSS of HW-MLP model

表 5　HW-MLP 模型与 SVM 算法对比

Tab. 5　Comparison betweenhw-MLP model and SVM model

对比内容	HW-MLP 模型	SVM 算法
训练时间	277.4s	449.6s
训练精度	98.00%	93.14%
测试精度	98.16%	79.19%
恢复密钥字节所需曲线条数	4.045	5.113

恢复其他密钥字节的训练精度、LOSS、测试精度和曲线条数如表 6 所示，经计算，平均训练精度为 97.75%，平均测试精度为 96.13%，恢复密钥字节平均需要 4.185 条能量曲线。

表 6　各 S 盒实验数据和恢复密钥字节需要的能量曲线条数

Tab. 6　Experimental data of each S-box and the number of power trace required to recover key bytes

序号	训练精度/%	训练 LOSS/%	测试精度/%	曲线数量
S_box_00	98.00	6.70	97.11	4.045
S_box_01	97.60	7.40	92.48	4.407
S_box_02	97.10	8.60	96.74	4.204
S_box_03	97.90	6.60	96.43	4.237
S_box_04	97.20	8.60	96.90	4.144
S_box_05	97.90	6.50	96.60	4.146
S_box_06	97.60	7.20	96.64	4.172
S_box_07	98.40	5.10	93.02	4.317
S_box_08	97.80	7.00	96.81	4.136
S_box_09	98.30	5.50	96.19	4.135
S_box_10	97.40	7.60	95.65	4.195
S_box_11	98.30	5.40	97.39	4.101
S_box_12	97.60	7.50	96.16	4.192
S_box_13	98.00	6.20	97.08	4.156
S_box_14	97.20	8.80	96.31	4.210
S_box_15	97.70	7.20	96.53	4.158

使用 HW-MLP 模型对完整密钥进行恢复，共进行 1000 次实验，恢复完整密钥平均需要 6.46 条能量曲线，各条数出现的次数如图 10 所示。

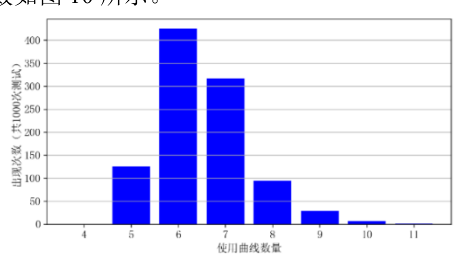


图 10　恢复 128bit 密钥需要的曲线条数分布

Fig. 10　Trace number distribution required to recover 128bit key

使用 HW-MLP 模型虽然增加了恢复密钥所需要的能量曲线条数，但是在由于模型结构的简化，训练时间和训练参数都有显著下降。

3.3.1 bit-MLP 模型实验

使用 bit-MLP 模型对密钥字节的 8 个 bit 进行恢复，以第 0 字节密钥的第 0 比特为例进行实验，模型在 epochs=200、batch_size=100 的条件下进行训练，最终训练精度达到

94.56%，测试精度达到 95.20%，如图 11 所示。对第 0 字节密钥的其他比特进行恢复，结果如表 7 所示，平均训练精度为 94.70%，测试精度为 93.39%，第 0 字节密钥恢复需要曲线条数为 1.469 条。

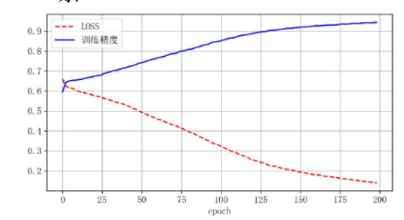


图 11　bit-MLP 模型训练精度与 LOSS

Fig. 11　Training accuracy and LOSS of bit-MLP model

表 7　第 0 个 S 盒 8 比特实验数据

Tab. 7　8-bit experimental data of the 0th S-box　/%

序号	训练精度	训练 LOSS	测试精度
bit_0	94.56	13.70	95.20
bit_1	95.23	12.37	95.55
bit_2	93.14	17.10	93.25
bit_3	94.08	15.05	94.22
bit_4	95.36	12.05	93.35
bit_5	95.43	11.67	88.40
bit_6	94.89	13.19	91.85
bit_7	94.95	12.82	95.32

恢复其他密钥字节的训练精度、LOSS、测试精度和曲线条数如表 8 所示，经计算，平均训练精度为 93.64%，平均测试精度为 92.56%，恢复密钥字节平均需要 1.540 条能量曲线。

表 8　各 S 盒实验数据和恢复密钥字节需要的能量曲线条数

Tab. 8　Experimental data of each S-box and the number of power trace required to recover key bytes

序号	训练精度/%	训练 LOSS/%	测试精度/%	曲线数量
S_box_00	94.70	10.80	93.39	1.469
S_box_01	94.13	14.72	93.01	1.450
S_box_02	89.86	23.72	89.34	1.812
S_box_03	94.67	13.41	93.84	1.421
S_box_04	89.88	23.73	89.29	1.846
S_box_05	94.60	13.67	93.81	1.408
S_box_06	93.66	15.53	91.99	1.539
S_box_07	96.15	09.96	94.75	1.386
S_box_08	93.23	16.45	92.16	1.502
S_box_09	95.89	10.52	94.87	1.391
S_box_10	93.29	16.37	92.22	1.577
S_box_11	95.98	10.37	94.50	1.381
S_box_12	92.54	17.96	92.30	1.549
S_box_13	94.96	12.84	93.86	1.413
S_box_14	90.45	22.55	88.70	1.954
S_box_15	94.26	14.37	92.92	1.535

使用 bit-MLP 模型对完整密钥进行恢复，共进行 1000 次实验，恢复完整密钥平均需要 3.837 条能量曲线，各条数出现的次数如图 12 所示。

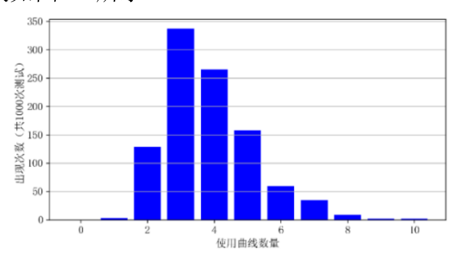


图 12　恢复 128bit 密钥需要的曲线条数分布

Fig. 12　Trace number distribution required to recover 128bit key

3.4　实验分析

1) 模型参数与性能对比

bit-MLP 模型和 ID-MLP 模型都能使用较少的能量曲线对密钥进行恢复，但是两种模型的结构和参数有较大差异，bit-MLP 模型结构简单，训练时间短，达到同样的训练精度(≥95%)所需要的训练集数量更少、时间更短，训练参数量比 ID-MLP 减少了 84%，训练时间减少 28%；HW-MLP 模型的训练时间和训练参数适中，但是在密钥恢复阶段需要的曲线条数较多。最后，为每个模型进行评分，模型恢复完整密钥需要的曲线条数记为 N，模型参数个数记为 P，模型的得分计算为 1/(N*P)，并将最高值按 100 分进行换算，各模型对比如表 9 所示。

因此，在实际的能量分析场景中，需要综合考虑能量曲线的输入维数、信噪比、采集的曲线条数和计算能力等情况对模型进行改进，选择最佳模型以达到最优性能。

表 9　各模型参数与性能对比

Tab. 9　Comparison of model parameters and performance

内容	Benadjila-MLP	ID-MLP	HW-MLP	bit-MLP
输出类别	256	256	9	2
模型层数	6	6	5	4
激活函数	relu	tanh	tanh	tanh
参数个数	270656	270656	76489	43642
单个 epoch 时间/s	1.5593	1.3546	1.3869	0.9753
训练集最少数量	50000	42000	32000	28000
测试精度	59.24%	95.51%	96.13%	92.56%
恢复密钥字节曲线数量	1.688	1.050	4.185	1.540
恢复完整密钥曲线数量	2.700	1.563	6.460	3.837
得分值	27.34	47.23	33.89	100.00

1) 针对 MLP 模型的防护策略研究

目前针对分组密码算法的防护主要有两种策略，一是信息隐藏技术，主要包括随机插入伪操作、乱序操作、增加噪声等；二是掩码技术，对每个中间值都被称为“掩码”的随机数进行掩盖，从而避免信息泄漏，姜久兴等人[18]提出的低面积复杂度低熵掩码方案，可以有效应对基于偏移量的 CPA 攻击。对 ChipWhisperer 实验平台进行随机插入伪操作并采集数据，使用 bit-MLP 模型对第 0 个 S 盒的第 0 比特进行训练和测试，训练精度和损失如图 13 所示，可以看出插入伪操作后，训练精度上升较慢，最终测试精度仅为 64.12%，略高于随机猜测精度，因此该策略具有一定的防御能力。

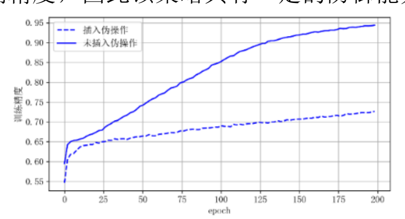


图 13　插入伪随机操作前后模型训练精度对比

Fig. 13　Comparison of training accuracy before and after inserting pseudo-random operation

4　结束语

将 MLP 神经网络引入到能量分析中，利用 MLP 神经网络提取特征的能力，可以挖掘出能量泄漏信息与所处理的敏感信息或密钥数据的深层次关系，为能量分析提供了一种新的手段和思路。实验结果表明，通过对文献[11]中的 MLP 模型进行改进，将神经网络结构进行优化，在减少了 84%的训练参数和 28%训练时间的情况下，可以有效提高测试精度，并减少密钥恢复阶段所需要的能量曲线条数，最少仅需要 1 条曲线即可完成 AES 算法 128 比特密钥的恢复。因此，本文

提出的 2 种模型具有训练参数较少、训练时间较短的优点，随着迭代次数的增加具有较高的预测精度和较强鲁棒性，该模型也能用于其他分组密码算法(如 DES、SM4 等)的侧信道能量分析和评估场景。

参考文献：

[1] Kocher P, Jaffe J, Jun B. Differential power analysis [C]// Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1999: 388-397.

[2] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems [C]// Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1996: 104-113.

[3] Gandolfi K, Mourtel C, Olivier F. Electromagnetic analysis: Concrete results [C]// International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, 2001: 251-261.

[4] De Mulder E, Buysschaert P, Ors S B, *et al.* Electromagnetic analysis attack on an FPGA implementation of an elliptic trace cryptosystem [C]// EUROCON 2005-The International Conference on"Computer as a Tool". IEEE, 2005, 2: 1879-1882.

[5] De Mulder E, Örs S B, Preneel B, *et al.* Differential power and electromagnetic attacks on a FPGA implementation of elliptic trace cryptosystems [J]. Computers & Electrical Engineering, 2007, 33 (5-6): 367-382.

[6] Liu Biao, Feng Huamin, Yuan Zheng, *et al.* Learning to attack from electromagnetic emanation [C]// 2012 6th Asia-Pacific Conference on Environmental Electromagnetics (CEEM) . IEEE, 2012: 202-205.

[7] Lerman L, Bontempi G, Markowitch O. A machine learning approach against a masked AES [J]. Journal of Cryptographic Engineering, 2015, 5 (2): 123-139.

[8] Maghrebi H, Portigliatti T, Prouff E. Breaking cryptographic implementations using deep learning techniques [C]// International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, Cham, 2016: 3-26.

[9] Gilmore R, Hanley N, O'Neill M. Neural network based attack on a masked implementation of AES [C]// 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) . IEEE, 2015: 106-111.

[10] Cagli E, Dumas C, Prouff E. Convolutional neural networks with data augmentation against jitter-based countermeasures [C]// International Conference on Cryptographic Hardware and Embedded Systems. Springer, Cham, 2017: 45-68.

[11] Benadjila R, Prouff E, Strullu R, *et al.* Study of deep learning techniques for side-channel analysis and introduction to ASCAD database [EB/OL]. ANSSI, France & CEA, LETI, MINATEC Campus, France. Online verfügbar unter https://eprint. iacr. org/2018/053. pdf.

[12] 冯登国, 周永彬, 刘继业等. 能量分析攻击 [M]. 北京: 科学出版社, 2010: 84-105. (Feng Dengguo, Zhou Yongbin, Liu Jiye, *et al.* Power Analysis Attack [M]. Beijing: science press, 2010: 84-105.)

[13] Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model [C]// International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2004: 16-29.

[14] Cagli E, Dumas C, Prouff E. Convolutional neural networks with data augmentation against jitter-based countermeasures [C]// International Conference on Cryptographic Hardware and Embedded Systems. Springer, Cham, 2017: 45-68.

[15] Bishop C M. Neural networks for pattern recognition [J]. Agricultural Engineering International the Cigr Journal of Scientific Research & Development Manuscript Pm, 1995, 12 (5): 1235-1242.

[16] Ruder S. An overview of gradient descent optimization algorithms [J]. arXiv preprint arXiv: 1609. 04747, 2016.

[17] Maghrebi H, Portigliatti T, Prouff E. Breaking cryptographic implementations using deep learning techniques [C]// International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, Cham, 2016: 3-26.

[18] 姜久兴, 厚娇, 黄海等. 低面积复杂度 AES 低熵掩码方案的研究 [J]. 通信学报, 2019 (5): 201-210. (Jiang Jiuxing, Hou Jiao, Huang Hai, *et al.* Research on area-efficient low-entropy masking scheme for AES [J]. Journal on Communications, 2019 (5): 201-210.)